Spectral Segmentation using Cartoon-Texture Decomposition and Inner Product-based metric

Wallace Casaca, Afonso Paiva, Luis Gustavo Nonato
Instituto de Ciências Matemáticas e de Computação – USP
{wallace,apneto,gnonato}@icmc.usp.br

***Abstract*—This paper presents a user-assisted image partition technique that combines cartoon-texture decomposition, inner product-based similarity metric, and spectral cut into a unified framework. The cartoon-texture decomposition is used to first split the image into textured and texture-free components, the latter being used to define a gradient-based inner-product function. An affinity graph is then derived and weights are assigned to its edges according to the inner product-based metric. Spectral cut is computed on the affinity graph so as to partition the image. The computational burden of the spectral cut is mitigated by a fine-to-coarse image representation process, which enables moderate size graphs that can be handled more efficiently. The partitioning can be steered by interactively by changing the weights of the graph through user strokes. Weights are updated by combining the texture component computed in the first stage of our pipeline and a recent harmonic analysis technique that captures waving patterns. Finally, a coarse-to-fine interpolation is applied in order to project the partition back onto the original image. The suitable performance of the proposed methodology is attested by comparisons against state-of-art spectral segmentation methods.**

***Keywords*-Spectral cut, image segmentation, similarity graph, cartoon-texture decomposition, harmonic analysis.**

I. Introduction

Image segmentation is no doubt one of the most important tasks in digital image processing and computer vision. The wide range of important applications that rely on image segmentation such as image coding, content-based image retrieval, and pattern recognition, have motivated the development of an enormous quantity of techniques for segmenting images. In particular, graph-based techniques figure among the most effective methods, mainly due to the flexibility it provides to handle color, texture, noise and specific features [1], [2], [3], [4], [5] in a unified framework.

The strength of graph-based approaches derive from the solid mathematical foundation it relies on, since most of the well-established graph theory [6] can be directly used to handle the image segmentation problem. For instance, spectral graph theory [7] has been the basic tool for the so-called spectral cuts method [1], [8], [9], [10], which exploits the eigenstructures of the image affinity graph so as to accomplish the image segmentation. In fact, spectral graph theory enables great flexibility in the segmentation process, as different choices can be made towards defining the similarity graph connectivity as well as the assignment of weights to the edges of the graph. Such a flexibility has leveraged a multitude techniques, turning out spectral cuts an attractive image segmentation approach.

Despite its flexibility and powerful, spectral cuts present some aspects that must be observed in order to ensure the success of the segmentation process. For example, the accuracy in detecting the boundaries between image regions is highly dependent on the weights assigned to the edges of the graph. Although automatic schemes have been proposed to compute those weights [1], [10], [11], [12], [13], it is well known that user intervention is essential in many cases to accurately define the object boundaries [14]. Therefore, incorporating user knowledge into the segmentation process is of paramount importance. Another important issue in the context of spectral cuts is the computational cost. Computing the eigenstructure of a graph is a very time consuming task, hampering the direct use of spectral cuts in high resolution images [15].

In this paper we present a new framework for image segmentation that relies on spectral cuts while addressing the issues raised above in an innovative manner. Firstly, we decompose the target image into two new images: the smooth and texture components. This mechanism is based on a cartoon-texture image decomposition scheme (Section III-A) that facilitates the identification of the different features contained in the image. Next, we provide a novel mechanism to assign weights to the edges of the affinity graph (Section III-C) that results in accurate segmentation in most cases. In contrast to other spectral cut-based approaches, our technique allows for user intervention, enabling to automatically modify weights according to the user perception (Section III-E). Moreover, we show how to build the similarity graph from a coarse representation of the input image without degrading segmentation results. Building the graph in a coarser resolution reduces the size of the graph, thus lessening the computational effort during the eigendecomposition, which permits to handle large images. Our results show (Section IV) that the proposed approach outperforms existing spectral image segmentation techniques in aspects such as accuracy and robustness.

Contributions We can summarize the main contributions of this work as:

- an image segmentation technique that combines cartoon-texture decomposition and spectral cuts;
- a novel method to compute and assign weights to the edges of a similarity graph using the cartoon component of the image;

- a new strategy to modify the weights of the graph according to user interaction, taking into account the texture component of the image.

II. Related Works and Basic Concepts

The literature on image segmentation is huge and a comprehensive overview about this theme is beyond the scope of this paper. In order to contextualize this work, we focus our discussion on methods based on graph partition to perform image segmentation.

Spectral methods: Given an image $\mathcal{I}$ it can be modeled as a weighted graph $G = (V, E, W)$, also called *pixel-affinity graph*, where each node $v_i \in V$ represents a pixel $P_i \in \mathcal{I}$, each edge $e_{ij} = v_i v_j \in E$ connects a pair of neighbor pixels and the weight $w_{ij} = w(e_{ij})$ is defined in terms of specific attributes such as luminance, position, and gradient in P_i and P_j. M. Fiedler, in his seminal work in graph theory [16], proposed a graph partitioning scheme based on spectral properties of the graph Laplacian matrix $L = (l_{ij})$, which is defined as:

$$l_{ij} = \begin{cases} -w_{ij}, & \text{if} \quad e_{ij} \in E \\ \sum_j w_{ij}, & \text{if} \quad v_i = v_j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Since L is symmetric and positive semi-definite, the solution of the eigenproblem

$$L\mathbf{x} = \lambda \mathbf{x}\,. \tag{2}$$

is given by non-negative real eigenvalues with respective real eigenvectors. The eigenvector associated to the second smallest eigenvalue of L is the so-called *Fiedler vector*. According to *Courant Nodal Domain theorem* [17], the zero-set of the Fiedler vector splits G into two disjoint graphs, thus the recursive computation of the Fiedler vector splits the graph hierarchically as a binary tree. This strategy has been used successfully in many approaches to segment images such as [18], [19], [20] (average cut) and [1] (spectral cut with unnormalized laplacian).

Normalized cut methods: Shi and Malik [1] introduced the concept of *Normalized Cut* (NCut), a graph optimization problem that aims to subdivide a graph so as to minimize the normalize sum of weights in the resulting graphs. They show that finding the optimal NCut is an NP-hard problem, but good solutions can be reached from the Fiedler vector computed by solving a generalized eigenvalue problem.

More specifically, the NCut method builds a graph G from a given image $\mathcal{I}$ by considering each pixel as a node of the graph and connecting two nodes v_i and v_j with an edge e_{ij} if

$$\|P_i - P_j\|_2 < r, \tag{3}$$

where P_i and P_j are the pixels associated to the nodes v_i and v_j while r is a parameter defining how local the edges should be. The weights $w_{ij} = w(e_{ij})$ assigned to the edges are computed from the following formula:

$$w_{ij} = \exp\left(-\frac{\|P_i - P_j\|_2^2}{\sigma_P^2} - \frac{\|I_i - I_j\|_2^2}{\sigma_I^2}\right), \tag{4}$$

where I_i and I_j account for intensity values in P_i and P_j. The pair (σ_P, σ_I) are tuning parameters used to control the scope of each term (position and intensity) in the edge weights.

Many variants of the NCut method have appeared in the literature, most of which proposing alternative graph construction and weight assignment. For instance, nodes of the graph and weights can be defined from watershed segmentation [21], [22], quad-tree decomposition [23], Markov random fields [24], texture descriptors [25] and biased normalized cuts [10], just to cite a few.

Multiscale methods: Aiming at achieving multiscale image segmentation, some authors have proposed to represent the graph in different levels, varying progressively the number of nodes and edges. The multilevel representation can be accomplished using conventional multiscale approaches based on a combination of graph compression and cross-scale constraint [11], shape information with PCA [13], and texture descriptors [26].

Other graph-based methods: Methods that relies on graph structures while avoiding spectral decomposition have also been proposed. *Image Foresting Transform* (IFT) [3], [27] is a good example of graph-based image segmentation technique that does not make use of spectral analysis. In short words, the IFT accomplishes the image partitioning by finding paths of minimum cost between seed nodes. IFT also allows user intervention to tune edge weights towards improving the segmentation results. A method based on isoperimetric regions [28] to find the shape with minimal perimeter has also been proposed successfully in the literature.

The technique describe in this paper proposes a new gradient-based weight computation which is only possible due to the cartoon-texture decomposition of the input image. Moreover, the proposed approach allows for user interaction so as to tune weights and thus improve the segmentation.

III. Pipeline overview

The proposed approach is comprised of five main steps, as illustrated in Fig. 1. The first step, *Cartoon-Texture Decomposition* decomposes the target image $\mathcal{I}$ into two images, $\mathcal{C}$ and $\mathcal{T}$, where $\mathcal{C}$ and $\mathcal{T}$ hold the cartoon and texture information contained in $\mathcal{I}$. In the second step an *image coarsening* is applied to $\mathcal{C}$ and $\mathcal{T}$. In the second step of the pipeline those images are coarsened so as to build smaller affinity graphs in the third step of the pipeline, namely the *affinity graph construction*. Besides speeding up the spectral decomposition, the reduced number of edges also lessen the computational burden during the weights assignment phase. Weights are derived from an inner-product-based metric defined on the coarse cartoon image. The spectral decomposition is carried out in the *spectral partition* step, being the result mapped back the original image through a coarse-to-fine interpolation procedure. The user can change the partition by stroking

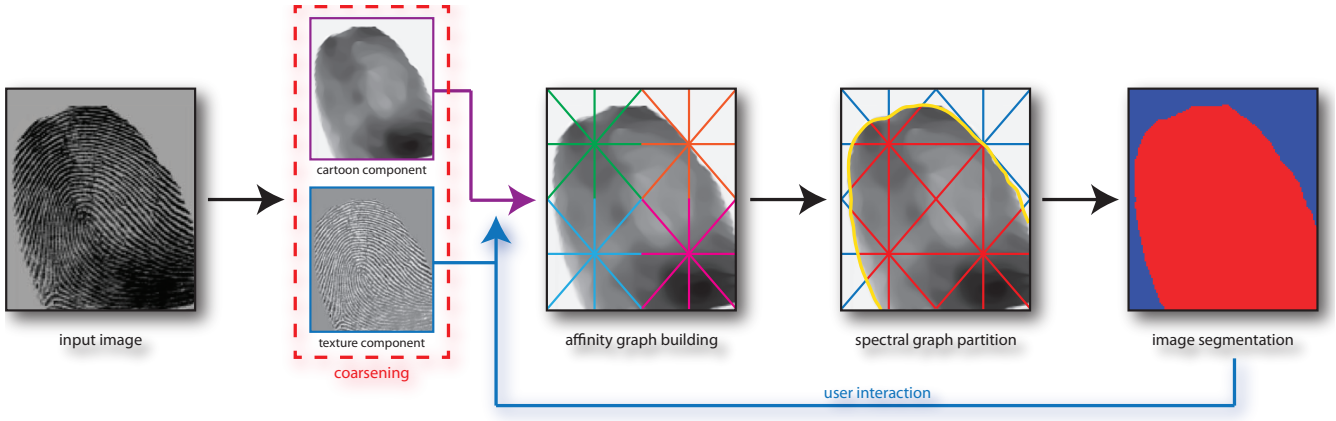


Fig. 1. Pipeline of our image segmentation framework.

the resulting segmentation. This step is performed by combining the coarse texture component with a recent technique of harmonic analysis.

Details about each step is presented below.

A. Cartoon-Texture Image Decomposition

Cartoon-Texture Decomposition (CTD) separate the input image $\mathcal{I}$ into two images, $\mathcal{C}$ and $\mathcal{T}$. The cartoon component $\mathcal{C}$ holds the geometric structures, isotopes and smooth-piece of $\mathcal{I}$ while the texture component contains textures, oscillating patterns, fine details and noise. Mathematically, these decomposition satisfies $\mathcal{I} = \mathcal{C} + \mathcal{T}$ (see [29], [30] and the underlying mathematical theory in [31]).

Both cartoon $\mathcal{C}$ and texture $\mathcal{T}$ components can be obtained by solving the following system of partial differential equations:

$$\begin{cases} \mathcal{C} = \mathcal{I} - \partial_x g_1 - \partial_y g_2 + \dfrac{1}{2\lambda}\,\mathrm{div}\left(\dfrac{\nabla \mathcal{C}}{|\nabla \mathcal{C}|}\right) \\ \mu \dfrac{g_1}{\sqrt{g_1^2 + g_2^2}} = 2\lambda \left[\dfrac{\partial}{\partial x}(\mathcal{C} - \mathcal{I}) + \partial^2_{xx} g_1 + \partial^2_{xy} g_2\right] \\ \mu \dfrac{g_2}{\sqrt{g_1^2 + g_2^2}} = 2\lambda \left[\dfrac{\partial}{\partial y}(\mathcal{C} - \mathcal{I}) + \partial^2_{xy} g_1 + \partial^2_{yy} g_2\right] \end{cases} \quad (5)$$

with initial conditions for $\mathcal{C}$, g_1, and g_2 given by

$$\begin{cases} \dfrac{\nabla \mathcal{C}}{|\nabla \mathcal{C}|} \cdot (n_x, n_y) = 0 \\ (\mathcal{I} - \mathcal{C} - \partial_x g_1 - \partial_y g_2) \cdot nx = 0 \\ (\mathcal{I} - \mathcal{C} - \partial_x g_1 - \partial_y g_2) \cdot ny = 0 \end{cases}, \quad (6)$$

Mathematically, the cartoon component $\mathcal{C}$ is a bounded variation function and the pair $\overrightarrow{g} = (g_1, g_2) \in L^2(\mathbb{R}^2)$ are such that the texture component $\mathcal{T} = \mathrm{div}(\overrightarrow{g})$. The constants $\lambda, \mu > 0$ are tuning parameters. Equations (5) are usually discretized by a semi-implicit finite difference schemes and solved using an iterative algorithm based on fixed point iteration [29]. Fig. 2 shows the result of the CTD scheme applied to an image.

In our context, both $\mathcal{C}$ and $\mathcal{T}$ are used to computed the weights assigned to the edges of the affinity graph. Since $\mathcal{C}$ is a texture-free denoised image, edge and shape detectors work well when applied to $\mathcal{C}$ [29]. This fact is exploited to define the weights, as we detail later. Information contained in $\mathcal{T}$ is handled only at the end of pipeline, during user interaction.

(a) input image (b) cartoon component (c) texture component

Fig. 2. Illustration of cartoon and texture decomposition.

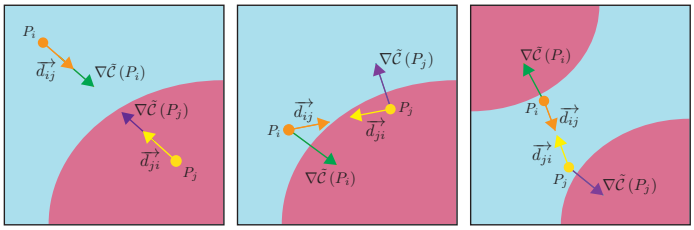
B. Image coarsening

In order to reduce the size of the affinity graph towards alleviating the computational burden during the spectral decomposition we perform a fine-to-coarse transformation on $\mathcal{C}$ (resp. $\mathcal{T}$), resulting in a coarse version $\tilde{\mathcal{C}}$ of $\mathcal{C}$. Such a transformation is accomplished using the bicubic interpolation method described in [32], which minimizes the blurring effect while still preserving gradients in the coarse image.

Our experiments showed that coarsening the image to one-fourth of its original resolution is a good trade-off between computational times and accuracy, speeding up the processing up to 6 times.

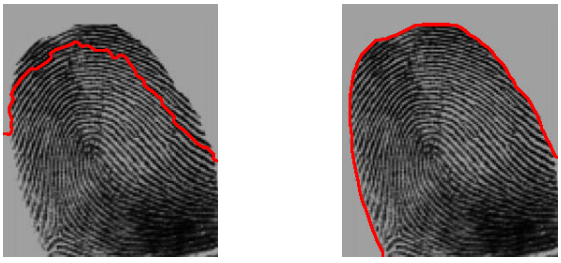
C. Building the affinity graph

The affinity graph G is built by associating each pixel from $\tilde{\mathcal{C}}$ to a node of the graph, connecting the nodes according to the distance between corresponding pixels (Eq. (3) with the supremum norm instead of euclidian). The weight assigned to each edge of G is derived from the proposed inner product-based metric. In contrast to the original NCut, which takes into account only spatial positions and pixel intensities (Eq. (4)), the inner product-based metric considers the variation of the image in the directions defined by the edges of the graph.



(a) Same direction (b) Oblique and orthogonal direction (c) Opposite direction

Fig. 3. Geometric interpretation of the inner product-based metric. Maximum weights occurs when the gradient and the direction defined from the graph edge point to the same direction (a). Moderate weight is highlighted in (b) and the third case, where opposite directions (c) produces minimum weights (zero).



(a) Without the CTD (b) The complete pipeline

Fig. 4. Automatic result with the proposed approach.

More specifically, the weight w_{ij} associated to the edge e_{ij} is defined as:

$$w_{ij} = \frac{1}{1+\eta g^2(i,j)}, \; g(i,j) = \max\left\{\frac{\partial\tilde{\mathcal{C}}(P_i)}{\partial\overrightarrow{d_{ij}}}, \frac{\partial\tilde{\mathcal{C}}(P_j)}{\partial\overrightarrow{d_{ji}}}, 0\right\}, \tag{7}$$

$$\frac{\partial\tilde{\mathcal{C}}(x)}{\partial\overrightarrow{d_{ij}}} = \nabla\tilde{\mathcal{C}}(x)\,\overrightarrow{d_{ij}}, \quad \overrightarrow{d_{ij}} = \frac{\overrightarrow{P_iP_j}}{|\overrightarrow{P_iP_j}|} \tag{8}$$

The left term in Eq. (8) is the directional derivative of $\tilde{\mathcal{C}}$ in the direction $\overrightarrow{d_{ij}}$, which is defined from the graph G. Therefore, image properties as well as the adjacency structure of the affinity graph is taken into account when assigning weights to the edges of G. In other words, similarly to Eq. (4), our formulation considers the intensity and geometric information to define the weights into a unique measure: the inner-product in the edges direction. Fig. 3 provides a geometric interpretation of the proposed inner product metric.

The effective weights w_{ij} are chosen from Eq. (7) rather then using the exponential measure usually employed by other authors [1], [23], [26]. The scheme proposed in (7) does not push values to zero as fast as the exponential function, which allows for considering the influence of a larger number of edges when carrying out the spectral decomposition. Eq (7) is indeed derived from the Malik-Perona diffusivity term [33], which was originally used for establishing the notion of anisotropy in the heat equation. Moreover, the inner product-based similarity metric (7) holds the property $w_{ij} = w_{ji}$, which ensures symmetry for the graph Laplacian matrix L. This fact is of paramount importance to guarantee that the eigenstructure of L is made up of only real numbers.

D. Spectral cutting and coarse-to-fine

Given the affinity graph G built from $\tilde{\mathcal{C}}$ and the number of partitions initially defined by the user we carry out the spectral decomposition using the same methodology proposed in [1]. More specifically, we first decompose the graph Laplacian matrix as $L = D - W$ where D and W contain the diagonal and off-diagonal elements of L (Eq. (1)). Then, the Fiedler vector f is obtained by solving the generalized eigenvalue problem

$$(D - W)\mathbf{x} = \lambda D\mathbf{x},$$

getting f as the eigenvector associated to the smallest non-zero eigenvalue.

The Fiedler vector splits $\tilde{\mathcal{C}}$ into two subsets, one containing the pixels corresponding to nodes of the graph where the entries of f are positive and other containing the pixels with negative values of f. Therefore, the zero-set of f is a curve that separates the regions with different signs. The partitioning obtained in $\tilde{\mathcal{C}}$ is brought back to $\mathcal{C}$ using bicubic interpolation from f. By recursively computing the spectral decomposition for each part of the image, one can produce finer segmentation. The recursive process can be driven by the user, who can specify the highest level of recursion, in addition to brushing defined by himself in any pieces of the image during each one of recursion steps.

Fig. 4 shows the result of applying our methodology to segment a fingerprint image. For the sake of comparison, we show in Fig. 4a the result of computing weights (using Eq. (7)) directly from the original image $\mathcal{I}$, that is, skipping the cartoon-texture decomposition. Notice from Fig. 4b how better the segmentation is when the CTD is employed.

E. Interactive Weight Manipulation

Weights can be interactively tuned so as to force the spectral cut to accurately fit boundaries between textured regions of the image. Our tuning scheme relies on the texture component $\mathcal{T}$ obtained from the cartoon-texture decomposition. The component $\mathcal{T}$ is processed by an harmonic analysis tool [34], [35] called *wave atoms*, which, in short words, assigns a scalar $S(\mathcal{T}_i) \in [0,1]$ to each pixel $\mathcal{T}_i$ of $\mathcal{T}$, where values close to 1 means the pixel belongs to the "wave" of a texture pattern, similar to that used in [36], [37]. Therefore, pixels nearby the boundary between two textured regions tend to be identified as not belonging to a texture wave, thus assuming values close to zero.

Starting from this premise, the weights of edges incident to pixels brushed by the user are modified as follows:

$$w_{ij} = k * \min_{e_{ij}\in E} w_{ij} * \max(S(\tilde{\mathcal{T}}_i), S(\tilde{\mathcal{T}}_j)), \tag{9}$$

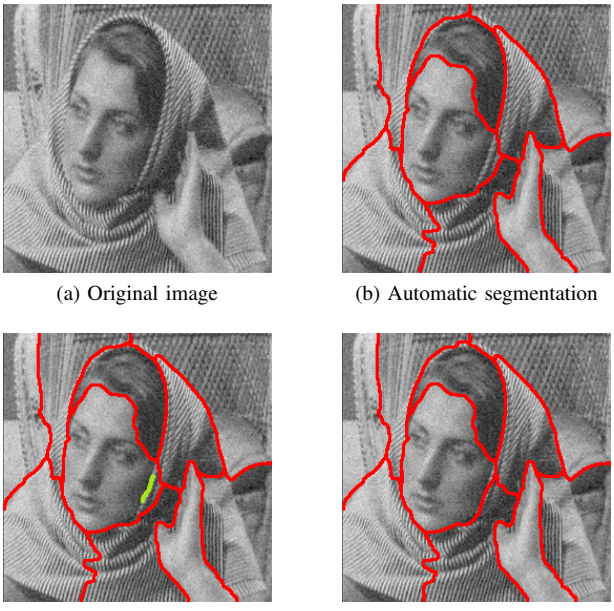
where $k \in (0, 1)$ is the smallest non-zero weight of the edges in G and $\tilde{\mathcal{T}}$ is the coarse version of $\mathcal{T}$, which was generated in Section III-A. The constant k enforces a more drastic change of weights in the region stroked by the user.

IV. Results and Comparisons

The following parameters were used in all experiments presented in this section: $\lambda = 0.05$ and $\mu = 0.1$ in the cartoon-texture decomposition (Section III-A), the default parameters suggested in [32] for the bicubic interpolation (Section III-B and Section III-D) and a hard threshold at 3σ (noise-to-signal ratio of the image) combined with cycle spinning [34] for the wave atom transform (Section III-E). We set $r = 1$ and $\eta = 5$ in equations (3)-(7), respectively. Finally, we have used parameters and implementations suggested by the authors for the methods used in our comparisons: k-way NCut [1] and multiscale NCut [11] (with radius $2, 3$ and 7 for each scale, respectively). Color images are converted to grayscale in order to be processed by the techniques.

User intervention We start showing how user intervention can be used to fix imperfections in the segmentation process. Fig. 5 shows the result of segmenting Fig. 5a using our methodology for 10 partitions. Notice from Fig. 5b that most parts of the image is accurately segmented, attesting the accuracy of the proposed method for the case where the image contains texture and moderate gaussian noise. The spectral cut deviates from the correct boundary in just a few small regions which are easily fixed through user interaction, as depicted in Fig. 5c and d.

Fig. 6 shows that the user does not need to perform a large number of interactions to fix a bad segmentation. The red curve in Fig. 6b is the result produced by our method without user intervention for two partitions. The simple greenish stroke on the top of the dog's head is enough to enforce a more satisfactory segmentation, as shown in Fig. 6c.



(a) Original image (b) Automatic segmentation (c) Brush made by user (green) (d) Final result

Fig. 5. Improving segmentation of the noise-textured image from user's strokes.

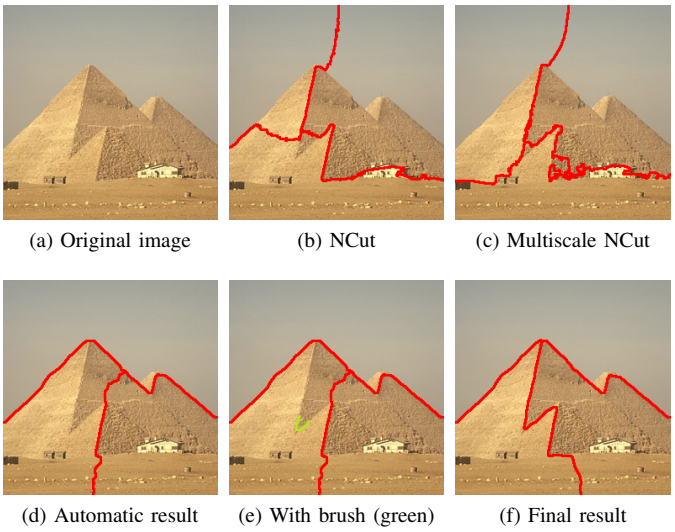
(a) Original image (b) Small stroke (c) Final result

Fig. 6. A simple stroke (greenish region on the dog's head) is enough to improve the segmentation.

Comparisons In order to confirm the quality of the proposed methodology we provide comparisons against two other techniques: the well known normalized cut [1] and the multiscale NCut technique [11], as mentioned earlier in this section. The first comparative test (Fig. 7) presents the result using user interference while all other results were performed through our automatic pipeline.

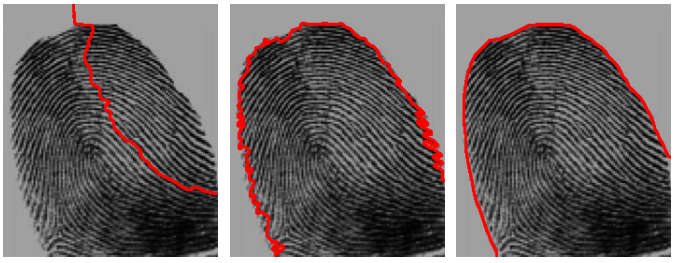
The experiment in Fig. 7 presents a comparative analysis of our technique against the other two approaches where user intervention is needed to improve the segmentation. We can see that both classical NCut (Fig. 7b) and multiscale NCut (Fig. 7c) badly segment parts of the image. Our approach results in a better partitioning (Fig. 7d), although some regions are also segmented in an incorrect way. After user intervention, shown in Fig. 7e, the result improves considerably (Fig. 7f).

The result of applying the three methods in a fingerprint image for two partitions is shown in Fig. 8. Notice that the NCut (Fig. 8a) does not segment the fingerprint correctly while



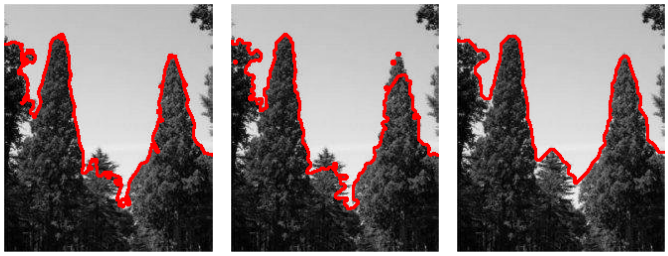
(a) Original image (b) NCut (c) Multiscale NCut (d) Automatic result (e) With brush (green) (f) Final result

Fig. 7. The influence of the user intervention in comparison with static approaches.



(a) NCut (b) Multiscale NCut (c) Ours

Fig. 8. The result of applying NCut, multiscale NCut, and our approach in a fingerprint image.



(a) NCut (b) Multiscale NCut (c) Ours

Fig. 9. Our approach produces smoother segmentation curves when compared to NCut and multiscale NCut.

the multiscale NCut and our approach do a good job. It is easy to see from Fig. 8b that the multiscale NCut tends to produce a jagged segmentation curve while our method results in a smoother curve, as shown in Fig. 8c.

It becomes clear from Fig. 9 that the smoothness of the result produced by our approach also help to increase robustness. While NCut and multiscale NCut tend to generate a segmentation curve with many artifacts and some cognition errors, our approach produces a much pleasant results.

Fig. 10 shows the partitioning produced by NCut, multiscale NCut, and our approach when applied to the images in the first column. These experimental images have been extracted from the *Berkeley Image Database* [38]. Fig. 10a, f, k, p and u show the input images wherein 5, $10, 15, 20$ and 25 partitions were computed for each technique, respectively. Notice that the multiscale NCut and our method produce much better results than the classical NCut (the ground truth is shown in the last column). In contrast to the multiscale NCut, our method produces smooth boundaries between the segmented regions, a characteristic also present in the ground truth images. Moreover, it can be seen that our method is more robust to identify structures contained in the images. For example, the man in the first row (Fig. 10d), the surfer (Fig. 10i), the soldier (Fig. 10s), and the stem of the mushroom (Fig. 10x) were better captured by our technique.

Computational effort The results presented in this paper were generated on a 1.80GHz AMD with 1GB of RAM. Table I describes some usage statistics recorded while producing the results. Our approach is considerably faster when compared against two other techniques. Based on our experiments and observations, the potential bottleneck in our framework is due to the Laplacian matrix assembly.

TABLE I
SOME STATISTICS AND TIMINGS (IN SECONDS) OF THE LAST EXPERIMENT.

image	*figure*	*NCut*	*NCut multiscale*	*Our approach*
man	Fig. 10a	95	18	8
surfer	Fig. 10f	200	30	17
bison	Fig. 10k	130	55	19
soldier	Fig. 10p	210	60	21
mushroom	Fig. 10u	240	60	21

V. Discussion and Limitation

The combination cartoon-texture decomposition and spectral cut turns out to be a quite efficient methodology for image segmentation. Moreover, the proposed inner product-based weight assignment mechanism has produced more accurate results than the exponential weighting function used by other spectral segmentation methods.

The comparisons presented in Section IV clearly show the effectiveness of the proposed spectral cut segmentation method, surpassing, in terms of accuracy, the state-of-art methods. Moreover, the flexibility as to user intervention is an important trait of our method, which enables the user to fix the segmentation locally.

There are two aspects to be observed when using our technique. First, the segmentation may not behave as expected if the user changes the weights substantially by stroking many parts of the image. It is worth noticing the this is a extreme case, since our method tend to produces quite satisfactory results without any user intervention. Another aspect to be considered is that the fine-to-coarse process may miss small details of the image.

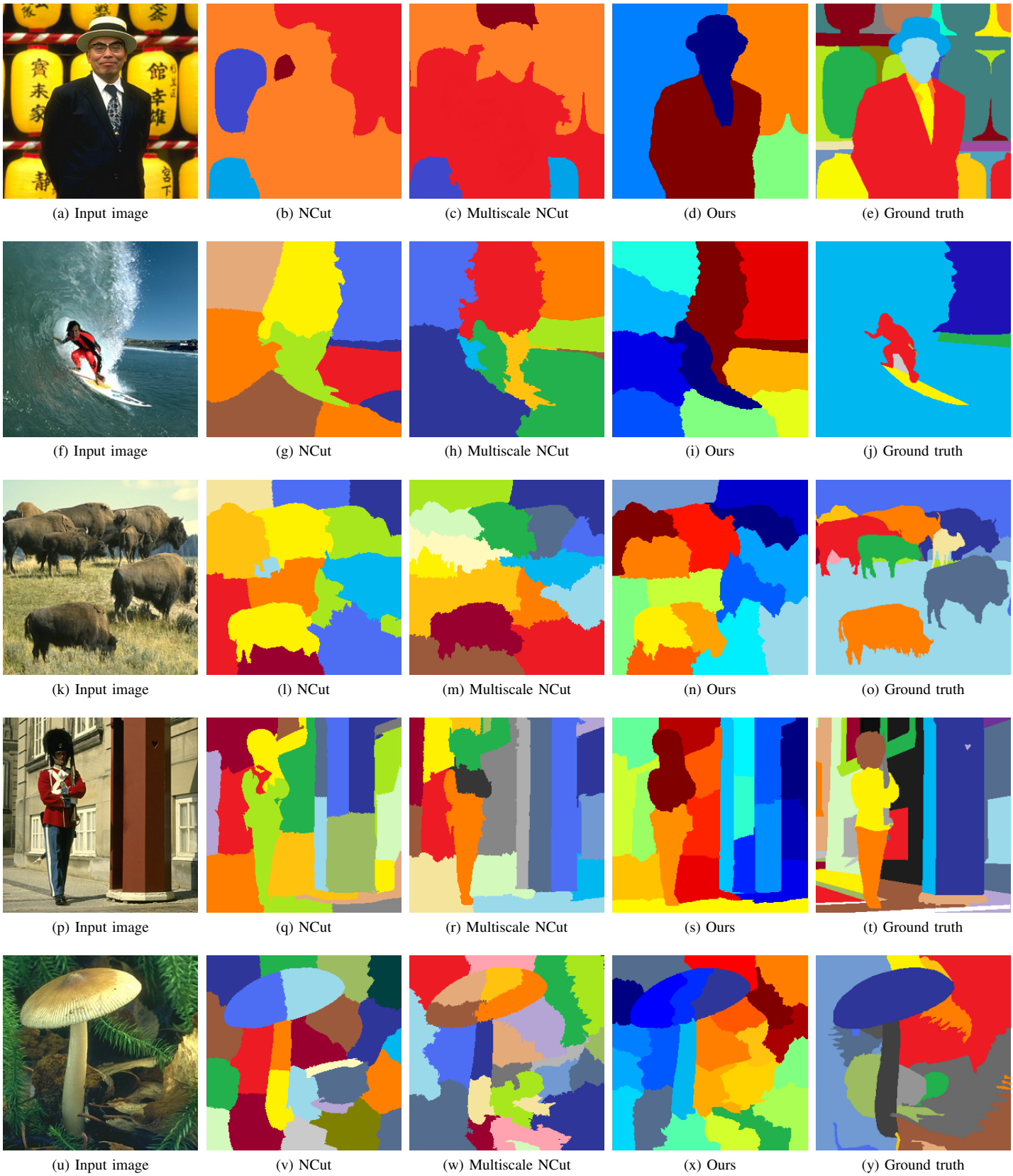
VI. Conclusion and Future Works

In this work we proposed a new methodology for spectral-cut-based image segmentation which relies on cartoon-texture decomposition. A new metric to measure the similarity between pixels and a new scheme to update weights of the affinity graph according to user intervention have also been presented. The evaluation we provided shows that our approach outperforms existing techniques in terms of accuracy and robustness, producing smoother segmentation curves.

We are currently investigating how to adapt the proposed methodology to become a truly multiscale method, in addition to evaluate the use of eigenvectors other than the first one. Moreover, we are also extending this methodology to 3D images in the context of medical data and to incorporate the texture component in an automatic pipeline for 2D images.

Acknowledgment

We would like to thank Jonatan F. Pereira and Rafael U. Nakanishi for their useful discussions and the anonymous reviewers for their constructive comments. This research has been funded by FAPESP-Brazil, CNPq-NSF, INCT-MACC and CNPq-Brazil.



(a) Input image (b) NCut (c) Multiscale NCut (d) Ours (e) Ground truth

(f) Input image (g) NCut (h) Multiscale NCut (i) Ours (j) Ground truth

(k) Input image (l) NCut (m) Multiscale NCut (n) Ours (o) Ground truth

(p) Input image (q) NCut (r) Multiscale NCut (s) Ours (t) Ground truth

(u) Input image (v) NCut (w) Multiscale NCut (x) Ours (y) Ground truth

Fig. 10. Partitioning produced by NCut, multiscale NCut, and our methodology. From top to bottom, the three methods partition the images in 5, 10, 15, 20 and 25 regions respectively.

REFERENCES

[1] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, 2000.

[2] J. Díaz, J. Petit, and M. Serna, "A survey of graph layout problems," *ACM Computing Survey*, vol. 34, pp. 313–356, 2002.

[3] A. Falcão, J. Stolfi, and R. Lotufo, "The image foresting transform: Theory, algorithms, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, pp. 19–29, 2004.

[4] W. Pratt, *Digital Image Processing (Chapter 17: Image segmentation)*, 4th ed. John Wiley & Sons, 2008.

[5] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 1, p. to te present, 2011.

[6] B. Bollobás, *Modern Graph Theory*. Springer, 1998.

[7] F. Chung, *Spectral graph theory*. CBMS, Reg. Conf. S. in Mathematics, Ameciran Mathematical Society, 1997.

[8] B. Mohar, "Some applications of laplacian eigenvalues of graphs," *Graph Symetric: Algebraic Methods and Applications*, vol. 497, pp. 225–275, 1997.

[9] D. A. Spielman, "Spectral graph theory and its applications," in *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society, 2007, pp. 29–38.

[10] S. Maji, N. Vishnhoi, and J. Malik, "Biased normlized cuts," in *Computer Vision and Pattern Recognition,CVPR*, 2011, p. to be presented.

[11] T. Cour, F. Bnzit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," in *In Proc. of CVPR*, 2005, pp. 1124–1131.

[12] D. Tolliver and G. L. Miller, "Graph partitioning by spectral rounding: Applications in image segmentation and clustering," in *Computer Vision and Pattern Recognition,CVPR*, vol. 1, 2006, pp. 1053–1060.

[13] W. Cai and A. C. S. Chung, "Shape-based image segmentation using normalized cuts," in *In Proceedings of ICIP'06*. IEEE Computer Society, 2006, pp. 1101–1104.

[14] L. Grady and A. K. Sinop, "Fast approximate random walker segmentation using eigenvector precomputation," in *Proc. of CVPR 2008*. IEEE Computer Society, June 2008.

[15] I. Koutis, G. L. Miller, and D. Tolliver, "Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing," in *International Symposium on Visual Computing*, 2009, pp. 1067–1078.

[16] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23(98), pp. 298–305, 1973.

[17] T. Biyikoglu, J. Leydold, and P. F. Stadler, *Laplacian eigenvectors of graphs: Perron-Frobenius and Faber-Krahn type theorems*. Springer, 2007.

[18] C. Jameson, A. Jujuunashvili, and A. Knyazev, "Modern eigenvalue solvers for spectral image segmentation," 2008. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.122.3624

[19] S. Sarkar and P. Soundararajan, "Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 504–525, 2000.

[20] P. Soundararajan and S. Sarkar, "Analysis of mincut, average cut and normalized cut measures," in *in Workshop on Perceptual Organization in Computer Vision*, 2001.

[21] W. Tao, H. Jin, and Y. Zhang, "Color image segmentation based on mean shift and normalized cuts," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, pp. 1382 –1389, 2007.

[22] M. Carvalho, A. Costa, A. Ferreira, and R. C. Junior, "Image segmentation using component tree and normalized cut," in *In Proceedings of Conference on Graphics, Patterns and Images(SIBGRAPI)*. IEEE Computer Society, 2010, pp. 317–322.

[23] M. Carvalho, A. Ferreira, and A. Costa, "Image segmentation using quadtree-based similarity graph and normalized cut," *Lectures Notes in Computer Science*, vol. 6419, pp. 329–337, 2010.

[24] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient n-d image segmentation," *Int. J. Comput. Vision*, vol. 70, pp. 109–131, 2006.

[25] X. Ma, W. Wan, and J. Yao, "Texture image segmentation on improved watershed and multiway spectral clustering," in *In Proceedings of ICALIP'08*. IEEE Computer Society, 2008, pp. 1693–1697.

[26] F. Sun and J.-P. He, "A normalized cuts based image segmentation method," in *In Proceedings of ICIC'09*. IEEE Computer Society, 2009, pp. 333–336.

[27] F. P. Bergo, A. X. Falcão, P. A. Miranda, and L. M. Rocha, "Automatic image segmentation by tree pruning," *J. Math. Imaging Vis.*, vol. 29, pp. 141–162, 2007.

[28] L. Grady and E. L. Schwartz, "Isoperimetric graph partitioning for image segmentation," *IEEE Trans. on Pat. Anal. and Mach. Int*, vol. 28, pp. 469–475, 2006.

[29] L. Vese and S. Osher, "Modeling textures with total variation minimization and oscillating patters in image processing," *Journal of Scientific Computing*, vol. 19, pp. 553–572, 2003.

[30] ——, "Color texture modeling and color image decomposition in a variational-pde approach," in *Proceedings of the Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*. IEEE Computer Society, 2006, pp. 103–110.

[31] Y. Meyer, *Oscillating patterns in image processing and nonlinear evolution equations*, 1st ed. University Lectures Series, Vol(22), American Mathematical Society, 2002.

[32] Y. Shuai, A. Masahide, T. Akira, and K. Masayuki, "High accuracy bicubic interpolation using image local features," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E90-A, pp. 1611–1615, 2007.

[33] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, pp. 629–639, 1990.

[34] L. Demanet and L. Ying, "Wave atoms and sparsity of oscillatory patterns," *Appl. Compt. Harmon. Anal.*, vol. 23, pp. 368–387, 2007.

[35] ——, "Curvelets and wave atoms for mirror-extended images," in *In Proceedings of SPIE Wavelts XII*, 2007, p. 67010J.

[36] W. Casaca and M. Boaventura, "A regularized nonlinear diffusion approach for texture image denoising," in *Proc. of the Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'09)*. IEEE Computer Society, 2009, pp. 164–171.

[37] ——, "A decomposition and noise removal method combining diffusion equation and wave atoms for textured images," *Mathematical Problems in Engineering*, vol. 2010, pp. 1–21, 2010.

[38] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, 2001, pp. 416–423.